

Comparison between Newton and response-surface methods

G.R. Kress and P. Ermanni

Abstract A supporting-point placement scheme is presented that is used for calculating function derivatives by the method of differences as well as a quadratic response-surface approximation. The placement scheme unifies the Newton (NM) and response-surface (RSM) methods in the limiting case when the point-set distance parameter for the RSM is chosen as small as that for obtaining the derivatives needed by the NM. Two new RSM minimization strategies with and without line searches are presented. The numerical performance of the algorithms is studied by using well-known test functions and the paths through the two-dimensional variables space are plotted for easier interpretation of the performance results. The results are compared with results of numerical experiments found in the literature.

Key words Newton method, response-surface method, sampling point placement schemes

1 Introduction

The computational effort for solving structural optimization procedures depends on the computing time for evaluating the structural model, the number of function evaluations, and the calculations performed by the minimization algorithm. For large structural models the numerical effort is dominated by the product of the structural-model solution time and the number of function evaluations. Thus, for a problem with given structural model, it is desirable to have a minimization algorithm that solves

the problem with as small a number of function evaluations as possible.

The hope to realize computational savings in solving large numerical optimization problems is at the heart of recent work, for instance by Canfield (2002) who uses a multi-point cubic approximation based on accumulating function and gradient information from multiple points during the iteration history of a sequential approximate optimization. Such methods can achieve a number of sampling points needed in each iteration less than that required by the Newton method or conventional RSM.

The focus of this research is a comparison of the Newton method and a basic RSM method using a new point placement scheme which achieves a smooth method transition just by changing the spacing of the point placement scheme. In addition, the placement scheme allows an RSM optimization strategy where function values of previous sampling points are inherited and the optimization process can be continued even when the minimum point of the approximation function corresponds to a higher value of the original function than the sampling points on which the approximation is based.

1.1 Characterization of the Newton and RSM methods

Mathematical programming provides efficient algorithms for convex and continuous objective functions with continuous derivatives. Both the NM and the RSM are derived by using properties of a quadratic polynomial. The NM is a second-order method because it requires both first and second function derivatives:

$$x^{(k+1)} = x^{(k)} - \mathbf{H}^{-1} \nabla f. \quad (1)$$

The first derivative is the gradient vector ∇f and the second derivative is the Hessian matrix \mathbf{H} .

On the other hand, the RSM does not require any derivatives. Both methods minimize a quadratic functional in one step. Assuming that derivatives are obtained by the differences method, the NM method and the RSM method considered here require the same number of function evaluations n_{fe} . Thus, the two methods are numer-

Received: 13 February 2003

Revised manuscript received: 21 April 2004

Published online: 5 July 2005

© Springer-Verlag 2005

G.R. Kress[✉] and P. Ermanni

Institute of Mechanical Systems, Structural Technologies,
Swiss Federal Institute of Technology, Leonhardstr. 27, 8092
Zurich, Switzerland

e-mail: gerald.kress@imes.mavt.ethz.ch

ically equivalent on quadratic functionals and the unification of the two methods through the supporting-points scheme presented here is discussed in this paper.

Practical optimization problems generally yield non-quadratic objective functions and both methods must seek the minimum points through iterations. Here, the two methods will generally perform differently because the NM bases its minimum-point estimates on information obtained in the small neighborhood of a reference point while the RSM method uses supporting points, the distance between which can span a wider region of the search space. However, if the distances between the supporting points for constructing the response-surface approximations are chosen as small as those for calculating derivatives, the RSM performs similarly to the NM.

Both methods have in common that, depending on the function landscape topology, the minimum-point estimates can be so poor that it is useful to enhance both basic algorithms by additional elements of a minimization strategy. The so-called modified Newton method combines the basic algorithm with a line search. Some minimization strategies elements for the RSM method, including the line search, are suggested in this paper.

1.2

Contents of this study

The methods are shortly recalled in Sect. 2. A scheme, placing supporting points in variables spaces of any dimension, is presented in Sect. 3. The points placed by the scheme determine the parameters of the quadratic response-surface approximation. If the distance between the points is chosen appropriately small, the point set is also well suited for obtaining the function derivatives needed for the NM. The evaluation scheme for obtaining the exact first and second derivatives of a quadratic polynomial is also explained. Common features and differences between the new point-placement scheme and the existing Latin hypercube design (LHD), which was first introduced by McKay *et al.* (1979), are outlined in Sect. 6.2.

A minimization strategy for the RSM is developed in Sect. 4. It allows the optimization process to continue even when the minimum-point estimates based on the response-surface approximation are occasionally divergent. Section 4.2 presents a modified RSM algorithm with line search.

The performance of the methods is studied on test functions that are well known from the literature. The minimization paths through the two-dimensional variables spaces are plotted, which helps to interpret the observed performance results.

In the literature, the conjugated gradient method by Fletcher and Reeves (2002) (FRM) is applied to the same test functions used in this study. The performance results reported by Reklaitis *et al.* (1983) are compared with those of NM and RSM obtained in this study.

Both methods offer the opportunity of taking advantage of parallel computing because the supporting points can be evaluated in parallel.

2

Methods recall

2.1

Second-order model: quadratic function

A quadratic function Π_2 is given by the form

$$\Pi_2 = p + \mathbf{x}^T \mathbf{p} + \mathbf{x}^T \mathbf{P} \mathbf{x}, \quad (2)$$

with the scalar p , the vector \mathbf{p} , and the matrix \mathbf{P} . As a necessary condition, the gradient must vanish at an extremum,

$$\nabla \Pi_2 = \mathbf{p} + 2\mathbf{P}\mathbf{x} = \mathbf{0}, \quad (3)$$

yielding the extreme point

$$\mathbf{x}_E = -\frac{1}{2}\mathbf{P}^{-1}\mathbf{p}. \quad (4)$$

The extreme point is a minimum when, for any vector \mathbf{v} with non-zero length, it holds:

$$\mathbf{v}^T \mathbf{P} \mathbf{v} > 0, \quad |\mathbf{v}| \neq \mathbf{0}. \quad (5)$$

2.2

Newton method

Consider a twice differentiable function f and its Taylor-series approximation \hat{f} up to the quadratic term:

$$\begin{aligned} \hat{f}(\mathbf{x}^{(k)} + \Delta \mathbf{x}) &= f(\mathbf{x}^{(k)}) + \Delta \mathbf{x}^T \nabla f(\mathbf{x}^{(k)}) + \\ &\frac{1}{2} \Delta \mathbf{x}^T \nabla \nabla f(\mathbf{x}^{(k)}) \Delta \mathbf{x}. \end{aligned} \quad (6)$$

The first and second derivatives of the approximation \hat{f} are:

$$\nabla \hat{f}(\mathbf{x}^{(k)} + \Delta \mathbf{x}) = \nabla f(\mathbf{x}^{(k)}) + \nabla \nabla f(\mathbf{x}^{(k)}) \Delta \mathbf{x}, \quad (7)$$

$$\nabla \nabla \hat{f}(\mathbf{x}^{(k)} + \Delta \mathbf{x}) = \nabla \nabla f(\mathbf{x}^{(k)}). \quad (8)$$

The extreme point condition is that the gradient vector vanishes, yielding:

$$\Delta \mathbf{x} = - \left[\nabla \nabla f(\mathbf{x}^{(k)}) \right]^{-1} \nabla f(\mathbf{x}^{(k)}). \quad (9)$$

Understanding $\Delta \mathbf{x}$ as the difference vector between the extreme point estimate \mathbf{x}^{k+1} and the reference point \mathbf{x}^k ,

$$\Delta \mathbf{x} = \mathbf{x}^{k+1} - \mathbf{x}^{(k)}, \quad (10)$$

yields the iteration rule

$$\mathbf{x}^{k+1} = \mathbf{x}^{(k)} - \left[\nabla \nabla f(\mathbf{x}^{(k)}) \right]^{-1} \nabla f(\mathbf{x}^{(k)}). \quad (11)$$

2.3

Response-surface method

The response-surface methodology was originally developed for constructing empirical response functions based on physical experiments. Replacing the physical experiments by numerical evaluations of a function allows the same methodology to find minimum points of that function. For that purpose quadratic polynomials are obviously suited. They are the simplest functions with a minimum point and can easily be constructed and evaluated. As stated by Venter (1998), response-surface approximations shift the computational burden from the optimization problem to the problem of constructing the approximations, and accommodate the use of detailed analysis techniques without the need for derivative information. Additionally, response-surface approximations filter out numerical noise inherent to most numerical analysis procedures by providing a smooth approximate response function, and simplify the integration of the analysis and the optimization codes.

Here, the response surface is approximated with a polynomial Π_2 . The following example is written out for two variables, or $n_x = 2$:

$$\Pi_2 = a_1 + a_2x_1 + a_3x_2 + a_4x_1^2 + a_5x_1x_2 + a_6x_2^2. \quad (12)$$

The coefficients c_i in terms of powers of the n_x optimization variables x_j, x_k can be obtained from the scheme given in Table 1. The parameters a_i of Π_2 can be adjusted to fit the objective function values at supporting points. The polynomial Π_2 (12) can be written as the inner product of the coefficient and parameter vectors, \mathbf{c} and \mathbf{a} :

$$\Pi_2 = \mathbf{c}^T \mathbf{a}. \quad (13)$$

Table 1 Coefficients \mathbf{c} for n_x supporting points

c_i	1	x_j	x_jx_k
i	1	$1+j$	$\frac{1}{2}j(2n_x-j+1)+k+1$
j		$1, n_x$	$1, n_x$
k			j, n_x

Table 2 Polynomial expression (2) in terms of (12)

	$p = a_1$	$p_j = a_i$	$\hat{P}_{jk} = \frac{1}{2}a_i,$	$\mathbf{P} = \hat{\mathbf{P}} + \hat{\mathbf{P}}^T$
i	1	$1+j$	$\frac{1}{2}j(2n_x-j+1)+k+1$	
j		$1, n_x$	$1, n_x$	
k			j, n_x	

We consider the case where the number of supporting points spanning the search space equals the number n_c of parameters or coefficients appearing in Π_2 . That number depends on the number n_x of optimization variables:

$$n_c = 1 + n_x + \frac{1}{2} [n_x(n_x + 1)], \quad n_x \geq 1. \quad (14)$$

Upon evaluation of n_c supporting points, the system of equations

$$\begin{bmatrix} 1 & c_{12} & \dots & c_{1n_c} \\ 1 & c_{22} & \dots & c_{2n_c} \\ 1 & \vdots & \ddots & \vdots \\ 1 & c_{n_c2} & \dots & c_{n_cn_c} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_c} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n_c} \end{bmatrix} \quad (15)$$

yields the approximation of the response surface with the unknown parameters \mathbf{a} .

The extreme point of the response-surface approximation is given by (4). The mapping of the parameters \mathbf{a} of the form (12) to the parameters p , \mathbf{p} , and \mathbf{P} of the form (2) is given in Table 2.

3

Placement of supporting points

Figure 1 shows the placement of the supporting points that can be used for a three-dimensional response-surface approximation. The point $\mathbf{x}^{(1)}$ at the center of the point set is the reference for constructing the other supporting points. Assuming the placement of the points on a regular lattice, indicated in Fig. 1, with a spacing D along

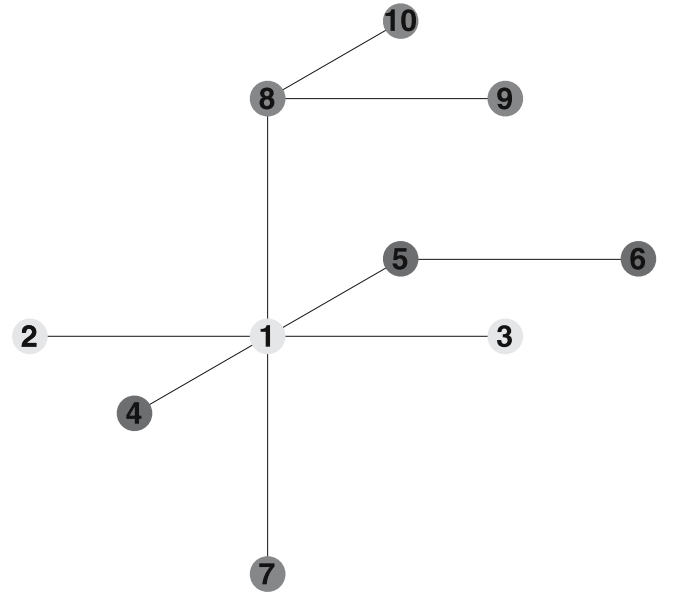


Fig. 1 Placement of supporting points $\mathbf{x}^{(1)}$ through $\mathbf{x}^{(10)}$ in three-dimensional variables space

Table 3 Supporting points in terms of changes with respect to $\mathbf{x}^{(1)}$

	2	3	4	5	6	7	8	9	10
1	$-D$	D	0	0	D	0	0	D	0
2	0	0	$-D$	D	D	0	0	0	D
3	0	0	0	0	0	$-D$	D	D	D

the respective coordinate directions, the coordinate variations of the individual supporting points with respect to the reference point $\mathbf{x}^{(1)}$ are given in the matrix presentation of Table 3. The rows correspond to the optimization variables and the columns correspond to the supporting points $\mathbf{x}^{(2)}$ through $\mathbf{x}^{(10)}$. The coordinates of the points can generally be identified for n_x variables by the following scheme. The points are identified successively from variable 1 through n_x . For the i^{th} variable, a number of $i+1$ supporting points must be added to the set of $n_c = n_c(i-1)$ (14) existing points. The first two of the new points span the direction of the i^{th} variable:

$$x_k^{(n_c+1)} = x_k^1 - \begin{cases} 0, & k \neq n_c + 1 \\ D, & k = n_c + 1 \end{cases} \quad (16)$$

and

$$x_k^{(n_c+2)} = x_k^1 + \begin{cases} 0, & k \neq n_c + 1 \\ D, & k = n_c + 1 \end{cases}. \quad (17)$$

The remaining $i-1$ points are variations of the new point $\mathbf{x}^{(n_c+2)}$ (17):

$$x_k^{(n_c+2+l)} = x_k^{(n_c+2)} + \begin{cases} 0, & k \neq l \\ D, & k = l \end{cases}, \quad l = 1, i-1 \quad (18)$$

The supporting points span some appropriate region of the search space. Along with the function values at each point, they are used to construct the response-surface approximation.

It is interesting to examine the presented supporting-point arrangement for the NM. The NM uses a set of supporting points in the vicinity of a reference point to calculate the gradient and Hessian matrix at that point. Point $\mathbf{x}^{(1)}$ then becomes the reference point at which the gradient and Hessian matrix must be obtained by the method of differences. The other supporting points must then move very close to the reference point, which is achieved by choosing a very small value for the distance D :

$$D = \epsilon \ll 1. \quad (19)$$

It is interesting to note that the point arrangement allows exact calculation of both first and second derivatives of quadratic polynomials, independent of the value of ϵ . This is illustrated in the three-dimensional case, where

the points 1 through 10 are arranged as shown in Fig. 1. Let the gradient and Hessian matrix be calculated from

$$\nabla f = \frac{1}{2\epsilon} \begin{pmatrix} f_3 - f_2 \\ f_5 - f_4 \\ f_8 - f_7 \end{pmatrix} \quad (20)$$

and

$$\nabla \nabla f = \frac{1}{\epsilon^2} (H_{ij}^a + H_{ij}^b + H_{ij}^c), \quad (21)$$

where \mathbf{H}^a is formed in terms of the reference point,

$$H_{ij}^a = - \begin{bmatrix} 2f_1 & -f_1 & -f_1 \\ -f_1 & 2f_1 & -f_1 \\ -f_1 & -f_1 & 2f_1 \end{bmatrix}, \quad (22)$$

\mathbf{H}^b is formed in terms of the points on the coordinate axes,

$$H_{ij}^b = \begin{bmatrix} (f_2 + f_3) & -(f_3 + f_5) & -(f_3 + f_8) \\ -(f_3 + f_5) & (f_4 + f_5) & -(f_5 + f_8) \\ -(f_3 + f_8) & -(f_5 + f_8) & (f_7 + f_8) \end{bmatrix}, \quad (23)$$

and \mathbf{H}^c is formed in terms of the points shifted away from the coordinate axes in other directions:

$$H_{ij}^c = \begin{bmatrix} 0 & f_6 & f_9 \\ f_6 & 0 & f_{10} \\ f_9 & f_{10} & 0 \end{bmatrix}. \quad (24)$$

Obviously, the gradient of quadratic functions is calculated exactly because (20) is the central differences method. It can be seen from (22), (23), and (24) that (21) produces a symmetric Hessian matrix. Moreover, it can be shown that the evaluations reproduce the symmetric version of the matrix \mathbf{P} , multiplied by two:

$$\nabla \nabla f = (P_{ij} + P_{ji}) \quad (25)$$

The presented point arrangement and evaluation scheme calculate for quadratic functions $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and regardless of the value of the epsilon parameter not only the Hessian but also the gradient exactly.

The RSM evaluation scheme obtaining the approximation parameters p , \mathbf{p} , and \mathbf{P} , suggested by Venter (1998) and recalled in Sect. 2.3, can therefore be replaced by $p = f(\mathbf{x}^{(1)})$, $\mathbf{p} = \nabla f$, and $\mathbf{P} = \frac{1}{2} \nabla \nabla f$ where ∇f , and $\nabla \nabla f$ are obtained by (20) and (21), respectively, for arbitrary values of D . For quadratic functions, both methods obtain the same approximation parameters. Therefore, the evaluation scheme of the Newton method becomes identical with that of the RSM method at the limit of D approaching very small values $D \rightarrow \epsilon$.

The entries in the gradient vector, where (20) represents the example for three variables, are calculated by the general scheme

$$\nabla f_k = \frac{f_{n_c(k-1)+2} - f_{n_c(k-1)+1}}{2\epsilon}. \quad (26)$$

The entries of the matrices \mathbf{H}^a , \mathbf{H}^b , and \mathbf{H}^c follow from

$$H_{ij}^a = - \begin{cases} 2f_1, & j = i \\ -f_1, & j \neq i \end{cases}, \quad (27)$$

$$H_{ij}^b = \begin{cases} (f_{n_c(i-1)} + f_{n_c(j-1)}), & j = i \\ -(f_{n_c(i-1)+2} + f_{n_c(j-1)+2}), & j \neq i \end{cases}, \quad (28)$$

and

$$H_{ij}^c = \begin{cases} 0, & j = i \\ f_{n_c(j)-i+1}, & j \neq i \end{cases}. \quad (29)$$

4 Minimization strategies

4.1 Strategies without line search

The original Newton method without line search is given by (1) and is therefore not further discussed.

4.1.1 RSM minimization algorithms without line search

If the point defined by (4) has a smaller function value than any of the supporting points, it is used as the reference point for the next set of supporting points. The distance value D of the next supporting-point set depends on the distance between the current and next reference points:

$$D = |\mathbf{x}^{k+1} - \mathbf{x}^k|. \quad (30)$$

This is illustrated in Fig. 2. A current point set in two-dimensional variables space (solid circles) defines a response-surface approximation. Its minimum point \mathbf{x}_E is the encircled square. It becomes the reference $\mathbf{x}^{(1)}$ for the new set of supporting points that is marked with squares. The minimum point estimate of this set is indicated by the encircled triangle, defining the next point set. As can be seen from Fig. 2, the distance values D of new point sets are always chosen so that the region covered by the new set extends to the previous reference point. Thus, if the successful minimum point estimate is outside the region of the current set, the new set will span a larger region. If the minimum point is found inside the region of the current point set, the new region will automatically contract. Expanding the point set region makes

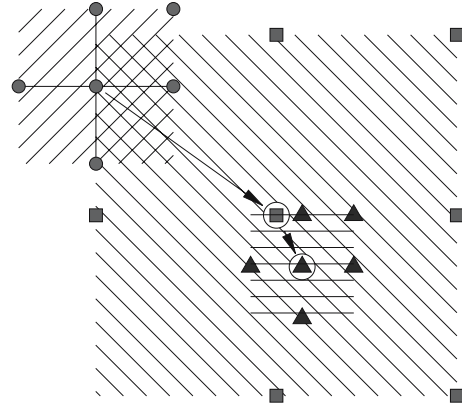


Fig. 2 Updated supporting-point sets around successful minimum-point estimates

it more likely to find the minimum point of the objective function inside that region when future minimum-point estimates will fail, and contracting the region helps accelerate convergence once it seems likely that the objective minimum point is being closed in.

If, on the other hand, the point defined by (4) does not yield a smaller function value than any of the supporting points, it is discarded and the strategy considers two cases.

In the first case, the current reference point $\mathbf{x}^{(1)}$ has a smaller function value than any of the other points $\mathbf{x}^{(i)}$, $i \neq 1$ of the set and remains the best estimate of the minimum point. Then the reference point $\mathbf{x}^{(1)}$ is kept and convergence is facilitated by shrinking the set of supporting points as illustrated in Fig. 3. In the present study a shrinking factor of $1/\sqrt{2}$ reduces the distance D .

In the second case, a point $\mathbf{x}^{(i)}$ other than the reference point $\mathbf{x}^{(1)}$ appears as the best minimum-point estimate. Then, the point set is shifted by the vector $\mathbf{u} = \mathbf{x}^{(i)} - \mathbf{x}^{(1)}$. The reference point of the shifted set coincides with the point with smallest function value of the previous set. As the spacing D is preserved, several of the other points will also coincide. Naturally, the function values of the coinciding points are already known. The numerical economy of this reminds one of the simplex search

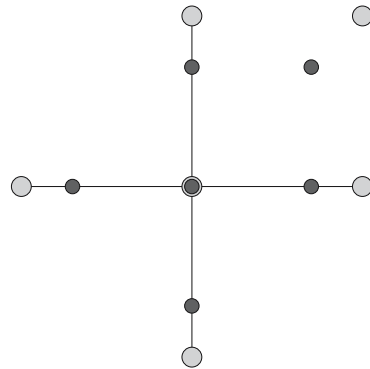


Fig. 3 Shrinking of supporting-point region around the reference point by a factor of $1/\sqrt{2}$

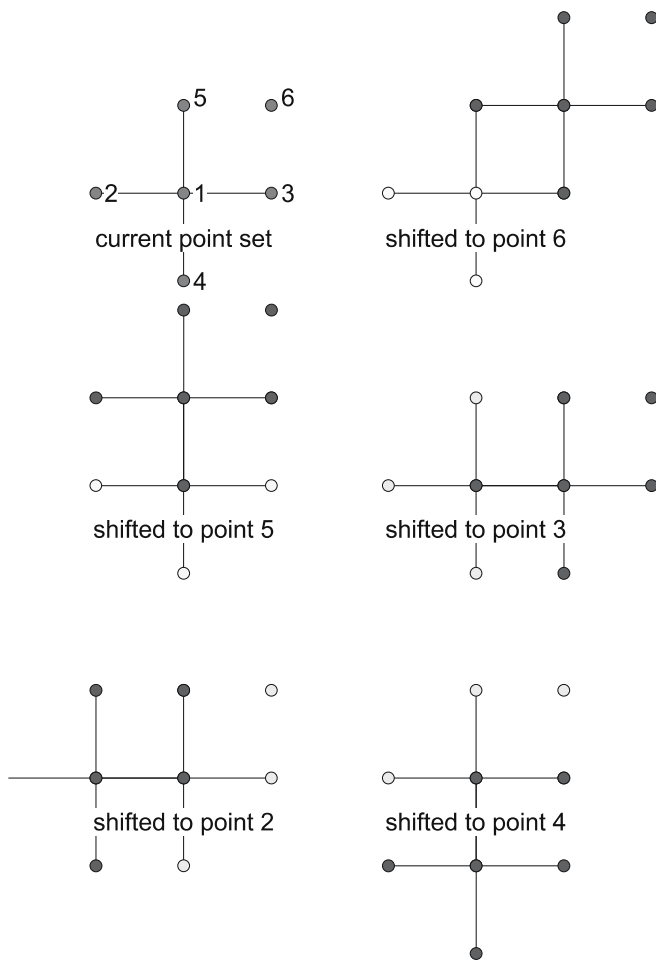


Fig. 4 Supporting-point set shifted to center around the respective points with smallest function value

method by Spendley *et al.* (1962). Figure 4 illustrates the shifts to the points two through six, respectively. The figure shows that in each case three points of the shifted sets coincide with previous points so that three function evaluations can be saved, making the optimization process more time efficient. In general, however, when moving the supporting-point set so that the reference point of the shifted set coincides with one of the other points in the original position, the total number of coinciding points depends on the point to which the reference point is shifted. Table 4 shows the number of coinciding points when the reference point of the new supporting-point set coincides with the other points of the previous set. In the one-dimensional variables space there are two other points where the reference point can be shifted and in each case there are two coinciding points. The situation in the two-dimensional variable space is illustrated in Fig. 4. In the three-dimensional variables space with ten supporting points the number of coincidences depends on the point of the previous set at which the reference point of the new set is located. The number of coincidences is either three or four. In the four-dimensional space the number of coincidences is either three or five. For a set of supporting points corresponding to n_x variables, the

Table 4 Number of coinciding points for the different reference point positions of the shifted sets corresponding to one-, two-, three-, and four-dimensional variable spaces

Shifted to point	No. of coinciding points			
	1D	2D	3D	4D
2	2	3	4	5
3	2	3	4	5
4		3	4	5
5		3	4	5
6		3	3	3
7			4	5
8			4	5
9			3	3
10			4	3
11				5
12				5
13				3
14				3
15				3

average number \bar{n}_c of coinciding points can be shown to be

$$\bar{n}_c = \frac{n_x}{n_p - 1} [2(n_x + 1) + 3(n_x - 1)] \quad (31)$$

The minimum, maximum, and average numbers of coinciding points, depending on the number of variables, are shown in Fig. 5. The maximum number increases linearly and the minimum number remains constant at three for variable spaces higher than one dimensional. The average number shown in the figure is calculated with (31). The points of each shifted set coinciding with previously evaluated points need not be evaluated again. Figure 6 shows the point set size according to (14) and the reduced size with minimum, average (31), and maximum number of coinciding points depending on the number of variables. The potential relative computing time savings

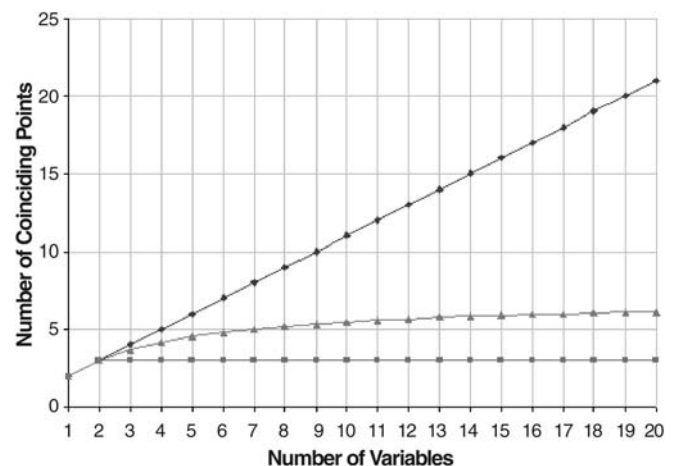


Fig. 5 Minimum, average, and maximum numbers of coinciding points depending on the number of variables

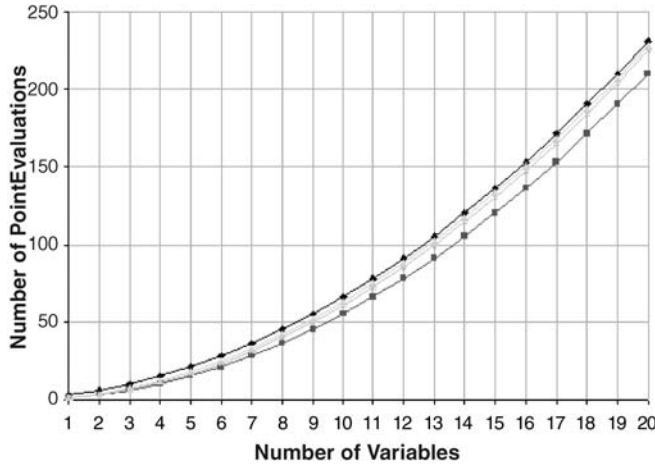


Fig. 6 Number of function evaluations versus number of variables and reductions due to point-shift mechanism

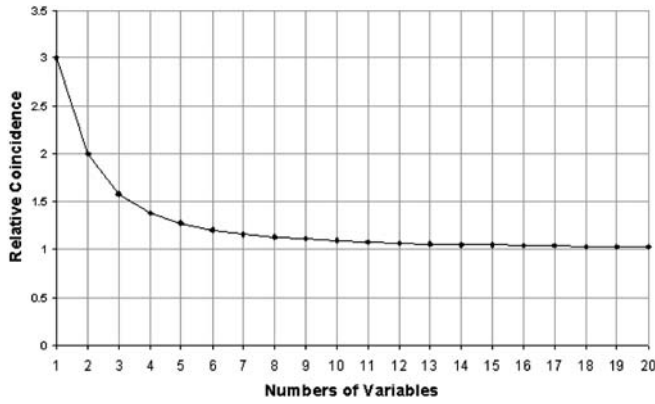


Fig. 7 Relative savings of function evaluations due to point-shift mechanism

are rather large when the number of variables is small but tend to become insignificant at large numbers of variables, as Fig. 7 illustrates. The figure plots the point-set size over the point-set size reduced by the average number of coinciding points,

$$e = \frac{n_p}{n_p - \bar{p}_c}, \quad (32)$$

which is a measure of the potential efficiency gain. As the figure shows, a program may run up to three times faster when searching for the minimum point of only one variable, or twice as fast when there are two variables, but only insignificantly faster when the number of variables is large.

4.2

Strategies with line search

A line search is an essential part of the method of steepest descent by Cauchy (1847), the method of feasible directions by Vanderplaats (1984), the zeroth-order method of conjugated directions by Powell (1964), the method of conjugated gradients by Fletcher and Reeves (2002), or

the quasi-Newton methods by Davidon (1959) and Powell (1963). All of these have in common that the minimization process is a sequence of line searches where a minimum is sought along the respective given or calculated search directions s .

Although line searches may require many iterations to obtain the minimum point with a specified accuracy, they may increase efficiency of even the Newton or the RSM methods if the functions are highly nonlinear.

Brent (1973) provides an efficient line search algorithm, called Brent's routine. It starts from a reference point $\alpha = 0$ with known function value and first establishes an upper bound α_u so that the minimum along the search direction is bracketed between the two positions. Then, it applies the golden section method to narrow the brackets around the minimum point. Finally, it relies on the quadratic approximation method to obtain the minimum point quickly with high precision.

4.2.1

Line search termination criteria

The line search routine used in the present study follows the principles of the line search algorithm of Brent's routine and is amended with an additional feature. The additional feature is the criterion used to switch from the golden section method to the quadratic approximation method. The criterion is derived from the concept that the deviation between the objective function and its quadratic approximation appears in the deviation between it and the cubic approximation. A cubic approximation can be obtained at virtually no cost from the four supporting points required by the golden section method while the three of these points with lowest function values establish a quadratic approximation. The golden section algorithm decreases the distance between the brackets enclosing the minimum point in steps. The accuracy of both the cubic and the quadratic approximations increases with decreasing distance between the brackets. This implies that some measure of the deviation between the two approximations will become smaller. Specifically, the deviation between the respective predictions of the minimum point yields the criterion upon whose fulfillment the algorithm switches from the golden section to the quadratic approximation methods.

The routine starts to rely on the quadratic approximation method if the estimated minimum points of both methods fall between the brackets established by the three points with smallest function values and if

$$\frac{\left(x_{\min}^{(quadratic)} - x_{\min}^{(cubic)}\right)}{\left(x_{\min}^{(quadratic)} + x_{\min}^{(cubic)}\right)} < 0.1. \quad (33)$$

The line search ends if the bracketed range and some measure of the quadratic deviation of the function values of the three supporting points from the mean value become smaller than an ϵ value:

$$|x_3 - x_1| < \epsilon \wedge \sum_{k=1}^3 (f_k - \bar{f})^2 < \epsilon, \quad \bar{f} = \frac{1}{3} \sum_{k=1}^3 f_k. \quad (34)$$

A value of $\epsilon = 5 \times 10^{-8}$ has been used for all the sample calculations presented in Sect. 5.

4.2.2 Modified Newton method

The modified Newton is well known in the literature:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^k \mathbf{H}^{-1} \nabla f \quad (35)$$

4.2.3 RSM minimization algorithms with line search

The original RSM minimization algorithm is enhanced by line searches.

When the estimated minimum point calculated by (4) has a lower function value than can be found in the supporting-point set, a line search is conducted. The search direction follows from the difference of the estimated minimum point \mathbf{x}_E and the reference point $\mathbf{x}^{(1)}$ of the supporting-point set:

$$\mathbf{s} = \mathbf{x}_E - \mathbf{x}^{(1)}. \quad (36)$$

When the estimated minimum point calculated by (4) does not have a lower function value than can be found in the supporting-point set, the shifting mechanism of the original method is enhanced by a line search. The search direction then follows from the difference of the minimum and reference points, $\mathbf{x}^{(i_{min})}$ and $\mathbf{x}^{(1)}$, of the supporting-point set:

$$\mathbf{s} = \mathbf{x}^{(i_{min})} - \mathbf{x}^{(1)}. \quad (37)$$

Of course, the line search annihilates the potential savings in function evaluations characteristic of the original shift mechanism.

5 Numerical examples

The performance of the following methods is studied:

1. Newton method (11),
2. modified Newton method (33),
3. response-surface method without line search, and
4. response-surface method with line search.

A value of $\Delta x = 10^{-6}$ is used for calculating derivatives by the forward differences method and an initial point-set spacing $D = 0.5$ is used for the RSM versions. Numerical experimentation indicated better performance values if the distance parameter is only allowed to remain constant or to decrease during the optimization sequence; the reported results are obtained with that rule.

5.1 Comparison with literature results

Reklaitis *et al.* (1983) present some numerical performance results. These are obtained with some conjugate-gradient and quasi-Newton methods using line searches. They consider various sample problems and the two-dimensional functions by Rosenbrock and by Eason and Fenton (1974) are also used in the present study. In order to reference the literature results presented here, the literature problems are also solved with the Fletcher–Reeves method and the results are compared with those presented by Reklaitis *et al.* (1983). The literature uses several line search algorithms, namely the interval halving method, the golden section method, Coggin’s method, and the cubic approximation method. The software used for this study includes an algorithm similar to Brent’s routine. Not allowing the switch to the quadratic approximation, the line search is conducted using the golden section method only. These results can be compared directly with those given in the literature and are listed in Table 5. Table 5 lists the number of iterations (here: line searches) n_{IT} , the number of function evaluations n_{FE} , and the achieved function values \bar{f}^* on termination. The literature does not mention the number of iterations. The present study obtains better efficiency than reported by Reklaitis *et al.* (1983). This may be due to the fact that the present study uses double-precision arithmetic while the literature results were obtained with single-precision arithmetic.

Reklaitis *et al.* (1983) obtain best results when the cubic approximation method is used for the line searches. Using Brent’s routine instead of the golden section method improves the results obtained in the present study. The improved results are listed in Table 6.

Table 5 Numerical results from (Reklaitis *et al.* 1983) (a) and present study (b) obtained with the golden section method

Problem	Ref.	n_{IT}	n_{FE}	\bar{f}^*
Rosenbrock	(a)	–	805	5.9×10^{-6}
	(b)	27	650	8.4×10^{-7}
Fenton/ Eason	(a)	–	199	1.744
	(b)	4	103	1.744

Table 6 Numerical results from (Reklaitis *et al.* 1983) (a) and present study (b) obtained with improved line search methods

Problem	Ref.	n_{IT}	n_{FE}	\bar{f}^*
Rosenbrock	(a)	–	273	2.8×10^{-7}
	(b)	27	348	4.3×10^{-6}
Fenton/ Eason	(a)	–	92	1.744
	(b)	4	62	1.744

5.2

Rosenbrock function

The Rosenbrock function is defined as

$$f = 100 (x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (38)$$

and illustrated by contour lines in Figs. 8, 9, 10, and 11. The contour lines do not present equidistant function values but allow for a better resolution of small function values: the function value differences increase, from line to line, by the fourth power. The starting point of the minimization procedures considered here is $\mathbf{x}^{(0)} = \{-1.2, 1.0\}^T$ and the minimum point is $\mathbf{x}^* = \{1.0, 1.0\}^T$ with function value $f(\mathbf{x}^*) = 0$.

The original Newton method solves the problem. Its solution path through the landscape of the Rosenbrock function is shown in Fig. 8. Instead of following the bot-

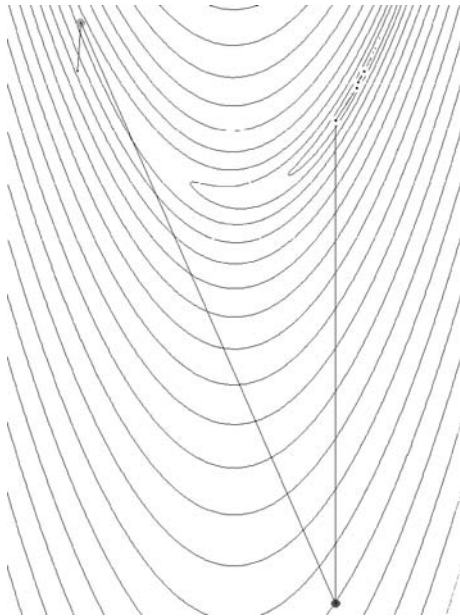


Fig. 8 Path taken by the Newton's method algorithm

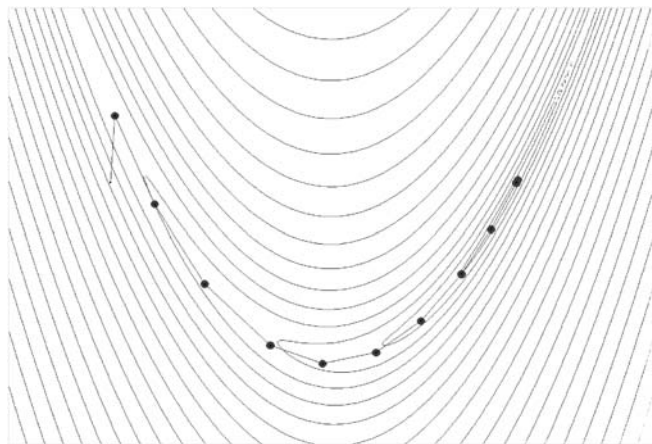


Fig. 9 Path taken by the modified Newton's method algorithm

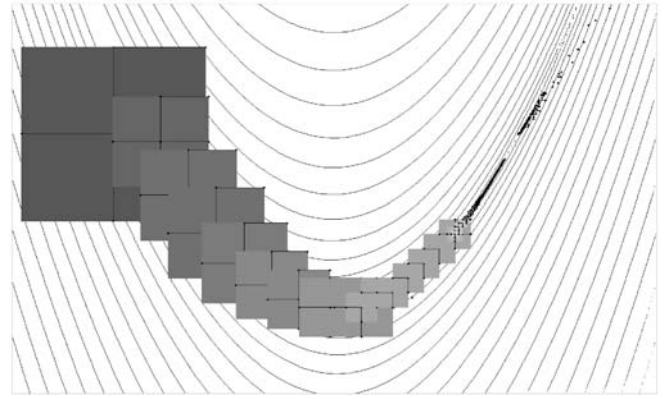


Fig. 10 Path taken by the RSM algorithm without line search

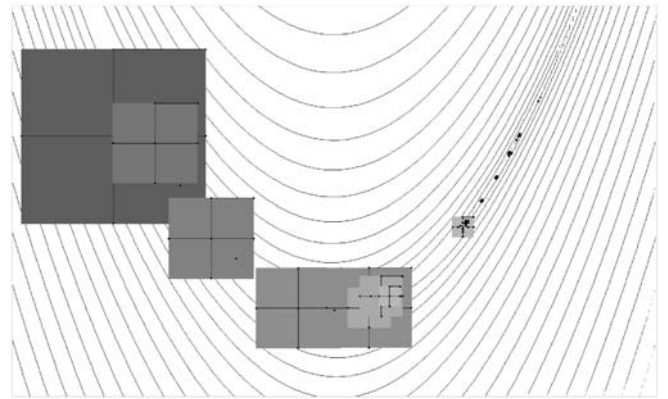


Fig. 11 Path taken by the RSM algorithm with line search and reference point based on minimum-point estimate

tom of the canyon, intermediate minimum-point estimates may occasionally be placed high up on the scarps. Nevertheless, the method performs better on the Rosenbrock problem than the other methods, as documented in Table 8.

Figure 9 illustrates the path along the bottom of the Rosenbrock function canyon taken by the modified Newton's method algorithm.

The performance of the original and modified Newton method is virtually reproduced by RSM with and without line search, if the spacing of the supporting-point set is restricted to $D \leq \Delta x$. The results listed in Table 7 agree closely with those of the Newton methods given in Table 8. The paths of these algorithms agree closely with those shown in Figs. 8 and 9. The results confirm that, in limiting cases, NM and RSM unify.

Table 7 Performance results for the Rosenbrock function with RSM algorithms mimicking Newton methods

Method	n_{IT}	n_{FE}	\bar{f}^*
RSM like Newton	8	55	4.0×10^{-9}
RSM like mod. Newton	13	235	7.1×10^{-10}

Table 8 Performance results for the Rosenbrock function

Method	n_{IT}	n_{FE}	\bar{f}^*
Newton	7	57	2.6×10^{-7}
Mod. Newton	13	242	4.9×10^{-8}
RSM no LS	179	741	3.6×10^{-8}
RSM with LS 1	25	348	7.7×10^{-10}

The globally spaced point sets of the RSM are depicted by filling the spanned area with grey levels proportional to the average function value $\bar{f} = \frac{1}{6} \sum_{k=1}^6 f_k$ that is, moreover, normalized with respect to the function value range contained in the plotted area of the Rosenbrock function. In addition, the individual supporting points of each set are indicated by dots, and lines are drawn between points \mathbf{x}^2 and \mathbf{x}^3 , \mathbf{x}^4 and \mathbf{x}^5 , and \mathbf{x}^5 and \mathbf{x}^6 . Each new point set is plotted on top of previous sets so that some sets are obscured from view.

The RSM algorithm without line search takes the path indicated in Fig. 10. The range from the top left to the middle of Fig. 11 indicates a sequence of successful minimum-point estimates where the values of $D^{(k)}$ tend to decrease. In the middle of the figure, another sequence starts dominated by the point-set shift mechanism with $D^{(k)}$ values remaining constant. The accumulation of dots in the top right of the figure indicates the unsuccessful minimum-point estimates activating the shifting mechanism.

The RSM algorithm version, where a line search replaces the shift mechanism, takes the path indicated in Fig. 11.

The performance characteristics of each method are listed in Table 8.

5.3

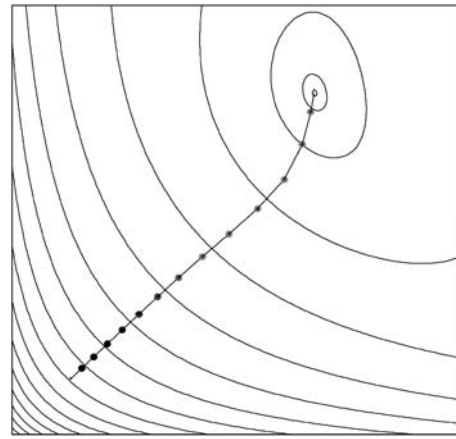
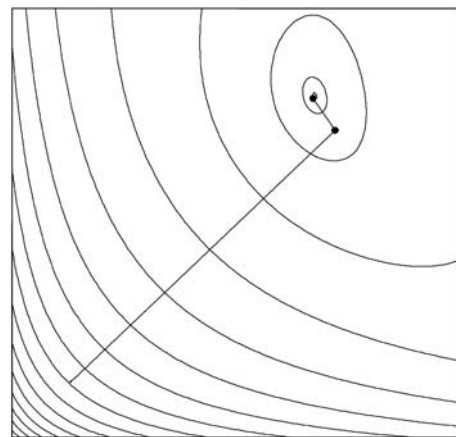
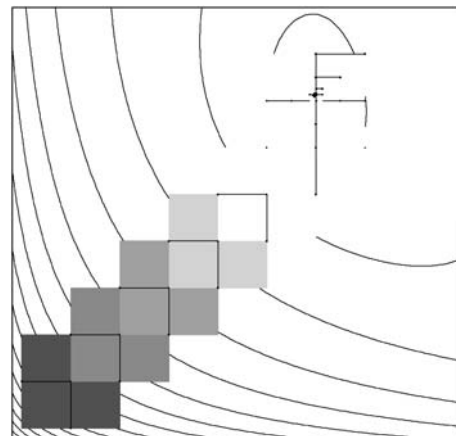
Fenton and Eason's function

The function used by Fenton and Eason is defined as

$$f = \frac{1}{10} \left(12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1^2 x_1^2)^4} \right) \quad (39)$$

and illustrated by the contour lines in Figs. 12, 13, 14, and 15. The function value increases with the fifteenth power of the order of contour lines. The starting point is $\mathbf{x}^{(0)} = \{0.5, 0.5\}^T$. The lines $x_1 = 0$ and $x_2 = 0$ are singular. Therefore, the initial and maximum size of the RSM supporting-point set is $D = 2.5$.

The original Newton method takes the path shown in Fig. 12. The estimated minimum points are always placed in useful directions but the distances from the reference point tend to be too short. This is due to the high deviation of the function from a quadratic polynomial. The distances from point to point tend to increase toward

**Fig. 12** Path taken by the Newton's method algorithm**Fig. 13** Path taken by the modified Newton's method algorithm**Fig. 14** Path taken by the RSM algorithm without line search

the minimum, where the function is better represented by its three-term Taylor-series approximation.

The modified Newton method takes the path shown in Fig. 13. The line search utilizes descent from the reference point to the other point along the search direction that is much closer to the minimum. The number of

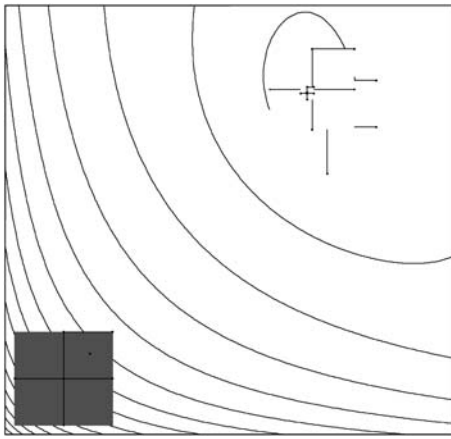


Fig. 15 Path taken by the RSM algorithm with line search

Table 9 Performance results for the function by Fenton and Eason with RSM algorithms mimicking Newton methods

Method	n_{IT}	n_{FE}	\bar{f}^*
RSM like Newton	17	119	1.7442
RSM like mod. Newton	5	105	1.7442

Table 10 Performance results for the function by Fenton and Eason

Method	n_{IT}	n_{FE}	\bar{f}^*
Newton	14	113	0.17442
Mod. Newton	4	68	0.17442
RSM no LS	12	65	0.17442
RSM with LS 1	5	127	0.17442

line searches until termination is only four and the process needs fewer function evaluations than the original method.

The performance of the original and modified Newton method is again approximated by RSM without line search and version 1 of the RSM, when the spacing of the supporting-point set is restricted to $D \leq \Delta x$. The results listed in Table 9 are similar to those of the Newton methods given in Table 10.

The RSM algorithm without line search takes the path shown in Fig. 14. It starts with a sequence of economic shifts, i.e. where only three function evaluations are needed to establish the new supporting-point set. This is followed by a few successful minimum point estimates and ends with rapid size reduction of the point set.

The RSM algorithm with line search takes the path shown in Fig. 15. The line search replaces the sequence of shifts so that the second supporting-point set is already much closer to the minimum than to the reference point. Nevertheless, the method needs more function evaluations than the version without line searches.

The performance characteristics of each method are listed in Table 10.

5.4

Discussion of numerical performance results

The numbers of function evaluations to obtain the minimum of the functions by Rosenbrock and by Fenton and Eason are summarized in Figs. 16 and 17, respectively. The horizontal bars in these figures indicate the number of function evaluations used by the conjugate-gradient method of Fletcher and Reeves (FRM). The results reveal that the relative performance of the various algorithms depends on the problem considered.

For the Rosenbrock function, the original and modified Newton methods outperform both FRM and RSM. The original NM needs even fewer function evaluations than its modified version. However, the path plotted in Fig. 8 reveals one diverging minimum-point estimate so that the favorable performance result seems to be effected by random chance rather than systematics. The RSM version with line search uses fewer function evaluations than the original one. Only the original RSM version is less efficient than the FRM.

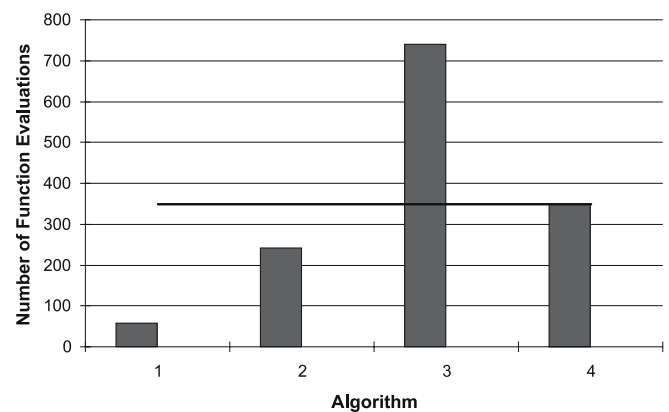


Fig. 16 Performance for the function by Rosenbrock. 1: Newton. 2: modified Newton. 3: RSM. 4: RSM with line search

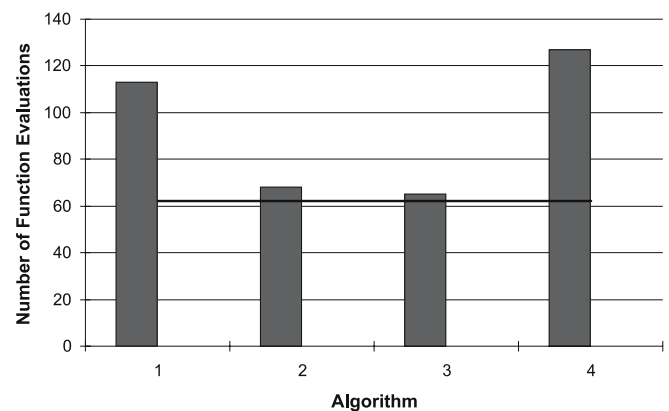


Fig. 17 Performance for the function by Fenton and Eason. 1: Newton. 2: modified Newton. 3: RSM. 4: RSM with line search

For the function by Fenton and Eason, on the other hand, FRM tends to perform better than the Newton methods and RSM. The line search improves the Newton method but slows down the RSM. The modified Newton method and original RSM nearly match the efficiency of the Fletcher–Reeves method.

6 Summary

The contributions of the present study are summarized in the following sections.

6.1 Relationship between the Newton method and RSM

The quadratic approximation of a response surface (objective function) requires a number of supporting points scaling with the number of optimization variables according to (14). The method of differences for numerically obtaining first and second derivatives of a function at a point also requires a set of supporting points with appropriately small separations.

The study presents, with (16) through (18), a placement scheme for supporting points in variable spaces of any dimension that can be used for both purposes. The evaluation of the placement scheme for the response-surface approximation and for obtaining first and second derivatives is also given with (26) and (27) through (29), respectively. On quadratic objective functions, the derivatives are calculated exactly regardless of the spacing between the points. This is because the point set, in any dimension, allows the gradient to be calculated by the central differences method while the Hessian matrix is already evaluated exactly by the forward differences method. The derivative information thus obtained can be used to construct the response surface in an alternative way.

NM and RSM find the minimum of a quadratic function exactly in one step. By using the same point placement and evaluation schemes presented here, they also require the same number of function evaluations. They are thus perfectly equivalent on quadratic functions depending on any number of variables.

Considering functions composed of higher-order polynomial terms, NM can be regarded as a limiting case of the RSM for $D \rightarrow \epsilon$, $\epsilon \ll 1$. This unification is effected by the particular supporting-point placement scheme presented. The theoretically identified similarity of both methods is numerically corroborated by applying the algorithms to test problems brought to the literature by Rosenbrock and by Eason and Fenton (1974). The NM and the limiting case of the RSM minimization algorithm outlined here solve the problems with almost identical numbers of function evaluations.

The similarity appears so interesting because the two methods are second and zero order, respectively, and

therefore often perceived as being of a fundamentally different nature.

6.2 Minimization algorithm based on RSM

The minimization algorithm, based on RSM and presented in this study, contains a shifting mechanism. It is used if the predicted minimum point has a higher function value than the smallest value of the supporting-point set. The shifting mechanism lets points of the new and previous supporting-point sets systematically coincide so that the number of new function evaluations is reduced.

Inheritance is also at the heart of RSM strategies, where the supporting-point sets are provided by LHD. LHD sampling can be controlled to yield a saturated set fulfilling the minimum requirement to fit a second-order model. Saturated LHD point sets and the new placement scheme have in common that the number of supporting, or sampling, points depends on the number of optimization variables by (14). Wang (2003) has used LHD to achieve an improved adaptive response-surface method (ARSM) with an increased probability of finding the global minimum of non-concave functions. It appears likely that LHD is better suited, for the design-space reduction strategies used by Wang, than the rigid lattice of the new point-placement scheme.

A version with line search, analogous to the modified Newton method, is also explained.

6.3 Numerical experiments

A link to numerical performance results given in the literature is established in Sect. 5.1. The results of this study are not only given in terms of numbers of function evaluations but the paths taken by the various algorithms through the landscapes of the two test problems considered are plotted so that the results can be better interpreted.

The original and modified Newton methods and the RSM-minimization algorithms presented in this study show similar performance, comparing well with that of the conjugated-gradient method by Fletcher and Reeves (2002).

7 Conclusion

The Newton and response-surface methods have been discussed and their common features have been exposed.

A particular supporting-point arrangement scheme is presented. It serves two purposes, constructing a quadratic approximation of response surfaces and obtaining the first two derivatives, equally well.

Along with the evaluation schemes presented in this paper, it provides identical scaling, in terms of number of function evaluations as a function of the number of optimization variables, for the Newton and response-surface methods when applied to the minimization of quadratic functions with unknown parameters.

An RSM-based algorithm for minimizing higher-order functions is presented where a point-set shift mechanism reduces the number of function evaluations.

The commonness of the Newton and RSM methods is also confirmed by the performance results obtained from the minimization of frequently used higher-order test problems.

The presented response-surface method minimization algorithms perform similarly to those based on the Newton and conjugated-gradient methods.

One of the advantages of RSM is their ability to filter out numerical noise. Apart from the wide spacing of the supporting points across the design space, a number of points higher than the number of model coefficients enhances the filter effect. Since the presented point-placement scheme produces exactly saturated point sets, the noise issue has not been addressed in this work.

Acknowledgements The project grant no. 2100-066879.01/1 of the Swiss National Science Foundation is gratefully acknowledged.

References

- Brent, R.P. 1973: *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall
- Canfield, R.A. 2002: Multi-Point Cubic Surrogate Functions for Sequential Approximate Optimization. AIAA Paper 2002-5575
- Cauchy, A. 1847: Method generale pour la resolution des systemes d'equations simultanees. *Comput Rend Acad Sci* **25**, 536–538
- Davidon, W.C. 1959: Variable Metric Method for Minimization. *AEC Res Dev Rep* **ANL-599**
- Eason, E.D.; Fenton, R.G. 1974: A Comparison of Numerical Optimization Methods for Engineering Design. *J Eng Ind Trans ASME* **96**(1), 196–200
- Fletcher, R.; Powell, M.J.D. 1963: A Rapidly Convergent Descent Method for Minimization. *Comput J* **6**, 163
- Fletcher, R.; Reeves, C.M. 1964: Function Minimization by Conjugate Gradients. *Comput J* **7**(5), 149–154
- McKay, M.D.; Bechmann, R.J.; Conover, W.J. 1979: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* **21**(2), 239–245
- Powell, M.J.D. 1964: An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *Comput J* **7**, 155–162
- Reklaitis, G.V.; Ravindran, A.; Ragsdell, K.M. 1983: *Engineering Optimization*. New Jersey: Wiley Interscience
- Spendley, W.; Hext, G.R.; Himsworth, F.R. 1962: Sequential Application of Simplex Designs in Optimization and Evolutionary Operation, *Technometrics* **4**, 441–461
- Vanderplaats, G.N. 1984: *Numerical Optimization Techniques for Engineering Design*. New York: McGraw-Hill
- Venter, G. 1998: Non-Dimensional Response Surfaces for Structural Optimization with Uncertainty. Dissertation, University of Florida
- Wang, G.G. 2003: Adaptive response Surface Method Using Inherited Latin Hypercube Design Points. *J Mech Des* **125**, 210–220